

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-345286

(43)Date of publication of application : 27.12.1993

(51)Int.Cl.

B25J 9/10

B25J 9/22

G05B 19/407

(21)Application number : 04-181576

(71)Applicant : FANUC LTD

(22)Date of filing : 17.06.1992

(72)Inventor : TERADA TOMOYUKI
UMIHORI KOJI

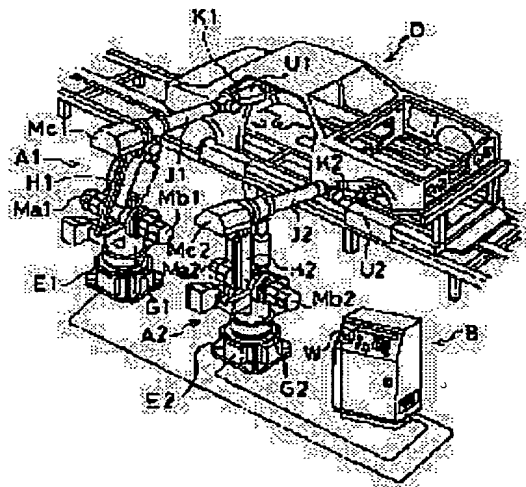
(54) METHOD FOR MANUAL FEEDING OF ROBOT

(57)Abstract:

PURPOSE: To constantly feed a robot manually at safe and efficient speed by automatically finding the shortest distance between the robots and manually feeding the robot by way of automatically lowering manual feeding speed as this shortest distance becomes shorter.

CONSTITUTION: In accordance with the current position and attitude of a robot A1 which is an object of taught operation and another robot A2, a control device B automatically finds the shortest distance between the robot A1 and the robot A2. Thereafter, by lowering feeding speed of the robot A1 which is to be the object of the taught operation as this shortest distance becomes shorter, manual feeding is carried out.

Consequently, even in the case when the distance between the robots A1 and A2 changes due to change of the position and attitude of the robots, it is needless for an operator to reset the feeding speed of the robot A1 which is to be the object of the taught operation, and additionally, the robot A1 is manually fed constantly at safe and efficient speed.



BEST AVAILABLE COPY

(19)日本国特許庁(JP)

(12)公開特許公報(A)

(11)特許出願公開番号

特開平5-345286

(43)公開日 平成5年(1993)12月27日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
B 2 5 J	9/10	A		
	9/22	Z		
G 0 5 B	19/407	E	9064-3H	

審査請求 未請求 請求項の数 5 (全 11 頁)

(21)出願番号 特願平4-181576

(22)出願日 平成4年(1992)6月17日

(71)出願人 390008235

ファナック株式会社

山梨県南都留郡忍野村忍草字古馬場3580番地

(72)発明者 寺田 知之

山梨県南都留郡忍野村忍草字古馬場3580番地 ファナック株式会社内

(72)発明者 海堀 弘次

山梨県南都留郡忍野村忍草字古馬場3580番地 ファナック株式会社内

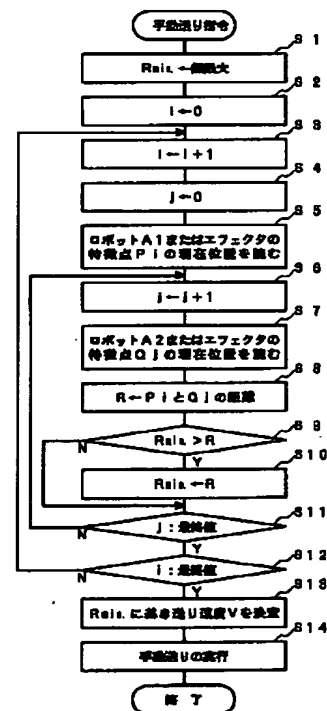
(74)代理人 弁理士 竹本 松司 (外2名)

(54)【発明の名称】 ロボットの自動送り方法

(57)【要約】

【目的】 多数のロボットを密集配備した場合であっても、各ロボットの教示操作を効率良く安全に行うことのできる自動送り方法を提供すること。

【構成】 コントローラBに自動送り指令が入力されている間、ロボットA1、A2に定義された特徴点Pi、Qjの現在位置座標を逐次シミュレーションしてRAMに更新記憶する。指標iおよびjの値を更新して特徴点Pi(i=1~最終値)と特徴点Qj(j=1~最終値)との間の距離Rを順次求め、より小さな特徴点間距離Rが算出される度にその値を最短距離記憶レジスタRminに更新記憶してロボットA1、A2間の最短距離を算出する(S1~S12)。最短距離Rminの値の大小に応じて実際の自動送り速度Vを増減することにより(S13~S14)、衝突発生の際のダメージを防止する一方、ロボット間距離に余裕があれば送り速度を速くして教示操作の効率化を図る。



【特許請求の範囲】

【請求項1】 相互に干渉する可能性のある位置に配備された複数のロボットを手動送りする方法において、ロボットの制御装置は手動送り指令の入力されたロボットと他のロボットの現在位置姿勢に基いてロボット間の最短距離を求め、前記最短距離が短いほど手動送り速度を低速にしてロボットを手動送りするようにしたロボットの手動送り方法。

【請求項2】 ロボットおよび該ロボットに装着したエフェクタ上に幾つかの特徴点を定義しておき、手動送り指令の入力されたロボットの前記特徴点の現在位置座標と他のロボットの前記特徴点の現在位置座標との間の距離を全て算出し、その最小値を求めて前記最短距離とすることを特徴とした請求項1記載のロボットの手動送り方法。

【請求項3】 ロボットの各リンクの形状および該ロボットに装着したエフェクタの形状の各々を単純形状のソリッドモデルもしくは単純形状のソリッドモデルの組み合わせに近似して形状データとして記憶しておき、手動送り指令の入力されたロボットのソリッドモデルの現在位置座標と他のロボットのソリッドモデルの現在位置座標を計算し、求められた位置座標によりロボット間の最短距離を求めることを特徴とした請求項1記載のロボットの手動送り方法。

【請求項4】 オペレータが指令した手動送り速度に、前記最短距離の減少に対応して単純減少する係数を乗じて実際の手動送り速度とすることを特徴とした請求項1、請求項2または請求項3記載のロボットの手動送り方法。

【請求項5】 前記最短距離が設定距離以下のときのみ実際の手動送り速度を指令手動送り速度よりも低速にすることを特徴とした請求項1、請求項2、請求項3または請求項4記載のロボットの手動送り方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ロボットの手動送り方法の改良に関する。

【0002】

【従来の技術】教示操作のためにロボットを手動送りする方法としては、教示操作盤等のオーバーライドキーを操作して送り速度を選択した後、各軸の手動送りキーによりロボットの各軸を個別に、または、連動させて駆動するものが知られており、また、教示操作で発生するロボットの衝突等に対処する安全策として、教示操作モードで選択できるオーバーライドの上限値を予め設定された許容値の範囲にクランプするものが知られている。そして、実際の教示操作では、教示操作の対象となるロボットの周辺に障害物が存在しないような場合においてはオーバーライド値を選択可能な範囲で大きく設定することにより手動送り速度を高速化し、また、教示操作の

対象となるロボットの近くに障害物が存在する場合、および、ロボットの手動送りによってロボットが障害物に接近したような場合では、オーバーライド値を小さく設定することにより手動送り速度を低速化して、教示操作の効率化と衝突発生時の安全対策とを同時に達成するようにしていた。

【0003】

【発明が解決しようとする課題】しかし、最近では作業空間の節約等のために多数のロボットを密集配備することが一般的に行われ、これに伴って、ロボットとロボットが相互に干渉するような状態で個々のロボットの教示操作を行う必要も生じてきた。このような場合、教示操作の対象となるロボットから見ると、その周辺に配備されているロボットは全て障害物となり得る。特に、複数のロボットを同時に教示操作するとき、あるロボットの教示操作を開始した後に一旦これを中断し、別のロボットの教示操作を行ってから再び前者のロボットの教示操作を再開したりするような場合では、前者のロボットの教示操作を一旦中断した時点における各ロボットの位置関係と再び前者のロボットの教示操作を再開した時点における各ロボットの位置関係、即ち、教示操作の対象となるロボットと障害物となるロボットとの距離が様々な変化する。従って、効率良く安全に教示操作を進めようとするれば、あるロボットの位置姿勢を変化させる毎に次の教示操作対象となるロボットのオーバーライド値を再設定しなければならないが、再設定操作自体が繁雑となるため、結果的に教示操作全体の作業効率が低下する場合がある。かといって、選択可能なオーバーライド値を低い値にクランプしてしまえば、教示操作の対象となるロボットの周辺に余裕がある場合でも高速の教示操作を行うことができず、また、周囲の状況を見逃して高速送りのためのオーバーライド値を選択したままの状態では教示操作を行えば、不用意な衝突を生じさせた場合のロボットのダメージが大きくなって危険である。

【0004】そこで、本発明の目的は、前記従来技術の欠点を解消し、多数のロボットを密集配備した場合であっても面倒な設定操作を必要とすることなく、各ロボットの教示操作を効率良く安全に行うことのできるロボットの手動送り方法を提供することにある。

【0005】

【課題を解決するための手段】本発明のロボットの手動送り方法は、ロボットの制御装置により手動送り指令の入力されたロボットと他のロボットの現在位置姿勢とに基いてロボット間の最短距離を自動的に求め、前記最短距離が短いほどロボットの制御装置により送り速度を自動的に低速にしてロボットを手動送りすることにより前記目的を達成した。また、求めた最短距離が設定距離以下のときにのみ実際の手動送り速度を指令手動送り速度よりも低速にすることにより、ロボット間の距離に余裕がある場合には高速の手動送り操作を行えるようにし

た。

【0006】

【作用】 教示操作等のための手動送り指令をロボットに入力すると、ロボットの制御装置は手動送り指令の入力されたロボットと他のロボットの現在位置姿勢に基いてロボット間の最短距離を求め、この最短距離が短いほど手動送り速度を低速にしてロボットを手動送りする。手動送り指令の入力されたロボット、即ち、教示操作の対象となっているロボットと他のロボットの現在位置姿勢に基いて制御装置が自動的にロボット間の最短距離を求め、この最短距離が短いほど教示操作の対象となるロボットの送り速度を低速にして手動送りを行うようにしたから、ロボットの位置姿勢変化によってロボット間の距離が変化した場合でも教示操作の対象となるロボットの送り速度をオペレータが再設定する必要がなく、しかも、常に安全かつ効率の良い速度でロボットが手動送りされるから、教示操作の高速化と安全性の確保が高い次元で両立される。

【0007】

【実施例】 以下、図面を参照して本発明の実施例を説明する。図1は2台の産業用ロボットA1、A2を相互に干渉する可能性のある領域に配備し、ライン上の自動車ボディDにスポット溶接を施す場合の例を示す斜視図である。産業用ロボットA1、A2の各々は、概略において、床面に固定されたベースE1、E2、ベースE1、E2に取り付けられて垂直軸の回りに回転するショルダーG1、G2、ショルダーG1、G2に取り付けられて垂直面内で揺動するアームH1、H2、アームH1、H2の先端に取り付けられて垂直面内で揺動するフォアアームJ1、J2、2つの回転自由度を有してフォアアームJ1、J2の先端に取り付けられ任意の姿勢変化が可能なリストK1、K2、および、各関節部を駆動するためのサーボモータMa1~Mc1、Ma2~Mc2等によって構成されている。ショルダーG1、G2、アームH1、H2、フォアアームJ1、J2、リストK1、K2の各々は各ロボットA1、A2のリンクであり、また、リストK1、K2の各々にはエフェクタとしてのスポット溶接器U1、U2が装着されている。

【0008】 各産業用ロボットA1、A2は単一の制御装置（以下、ロボットコントローラという）Bにより各々独立または協調して駆動制御されるようになっており、また、ロボットコントローラBには、産業用ロボットA1、A2に教示操作や手動送り操作を行ったりするための教示操作盤Wがコード接続されて着脱自在に載置されている。ロボットコントローラBは、制御手段としてのCPUや、産業用ロボットA1、A2のための駆動制御プログラムを格納したROM、および、教示データやシミュレーションデータ等を記憶するための不揮発性RAMや、ロボット接続用のインターフェイス等を備えた通常のロボットコントローラであって、その一般的な

構成や機能に関しては既に周知である。以下、前述の構成を参照して本発明の方法を適用した幾つかの実施例について説明する。

【0009】 まず、特徴点の現在位置座標に基いてロボット間の最短距離を算出することによって教示操作の対象となるロボットの手動送り速度を決定するようにした実施例について説明する。この場合、産業用ロボットA1の各リンクであるショルダーG1、アームH1、フォアアームJ1、リストK1、および、サーボモータMa1、Mb1、Mc1と、エフェクタであるスポット溶接器U1上に幾つかの特徴点Pi（i=1~任意の最終値）を定義し、また、産業用ロボットA2の各リンクであるショルダーG2、アームH2、フォアアームJ2、リストK2、および、サーボモータMa2、Mb2、Mc2と、エフェクタであるスポット溶接器U2上に幾つかの特徴点Qj（j=1~任意の最終値）を定義して、産業用ロボットA1、A2の形状データとしてロボットコントローラBの不揮発性RAMに記憶させておき、周知のシミュレーション機能により、産業用ロボットA1、A2の各々の現在位置姿勢に基いて各々の特徴点Pi、Qjの現在位置座標の全てを、例えば軸補間周期毎に算出し、ロボットコントローラBの不揮発性RAMに更新記憶させるようにしておく。なお、特徴点Pi、Qjを定義する位置は産業用ロボットA1、A2の各リンクやサーボモータ等の突出部材およびエフェクタの輪郭形状を端的に示す位置が望ましく、例えば、各リンクやサーボモータおよびエフェクタの両端部や両側部とする。

【0010】 図2は、オペレータが教示操作盤Wを操作して産業用ロボットA1またはA2に手動送り指令を入力している間にロボットコントローラBのCPUが所定周期毎に繰り返し実行する「手動送り処理」の概略を示すフローチャートであり、この処理のためのプログラムはロボットコントローラBのROMに駆動制御プログラムの一部として格納されている。

【0011】 教示操作盤Wからの手動送り指令を検出したCPUは、まず、最短距離記憶レジスタRmin.にシステムが許容する設定可能最大値を初期設定して検索指標iの値を“0”に初期化した後（ステップS1、ステップS2）、該指標iを“1”インクリメントして検索指標jの値を“0”に初期化する（ステップS3、ステップS4）。次いで、CPUは検索指標iの値に基いて産業用ロボットA1の第i番目の特徴点Piの現在位置座標を不揮発性RAMから読み込んで一時記憶すると共に検索指標jを“1”インクリメントし（ステップS5、ステップS6）、該指標jの値に基いて産業用ロボットA2の第j番目の特徴点Qjの現在位置座標を不揮発性RAMから読み込み（ステップS7）、特徴点PiおよびQjの現在位置座標に基いて特徴点PiとQjとの間の距離を算出して特徴点間距離記憶レジスタRに一時記

憶する(ステップS8)。

【0012】次いで、CPUは最短距離記憶レジスタRmin.の値と特徴点間距離記憶レジスタRの値の大小関係と比較し(ステップS9)、特徴点間距離記憶レジスタRの値が最短距離記憶レジスタRmin.の値よりも小さければ特徴点間距離記憶レジスタRの値を最短距離記憶レジスタRmin.に再設定し(ステップS10)、また、特徴点間距離記憶レジスタRの値が最短距離記憶レジスタRmin.の値よりも小さくなければ最短距離記憶レジスタRmin.の値をそのまま保持する。なお、この段階では最短距離記憶レジスタRmin.にシステムが許容する設定可能最大値が記憶されているから、ステップS9の判別結果は真となり、最短距離記憶レジスタRmin.に特徴点間距離記憶レジスタRの値、即ち、産業用ロボットA1の特徴点P1と産業用ロボットA2の特徴点Q1との間の距離が記憶されることとなる。

【0013】次いで、CPUは検索指標jの値が最終値に達したか否かを判別するが(ステップS11)、達していなければ、以下、ステップS11の判別結果が真となるまでの間、ステップS6～ステップS9とステップS11もしくはステップS6～ステップS11までのループ状の処理を繰り返し実行し、産業用ロボットA1の特徴点P1と産業用ロボットA2の特徴点Qj〔j=2～最終値〕との間の距離を全て算出し、最短距離記憶レジスタRmin.の現在値よりも小さな値の特徴点間距離が算出される毎に、その値を最短距離記憶レジスタRmin.に更新して保存する。このような処理を繰り返し実行する間にステップS11の判別結果が真となると、CPUは、検索指標iの値が最終値に達したか否かを判別するが(ステップS12)、達していなければ、以下、ステップS12の判別結果が真となるまでの間、前述のループ状の処理とステップS12の判別処理を繰り返し実行し、産業用ロボットA1の特徴点Pi〔i=2～最終値〕と産業用ロボットA2の特徴点Qj〔j=1～最終値〕との間の距離を全て算出し、最短距離記憶レジスタRmin.の現在値よりも小さな値の特徴点間距離が算出される毎に、その値を最短距離記憶レジスタRmin.に更新して保存する。

【0014】従って、ステップS12の判別結果が真となった段階では、産業用ロボットA1およびスポット溶接器U1の特徴点Pi〔i=1～最終値〕と産業用ロボットA2およびスポット溶接器U2の特徴点Qj〔j=1～最終値〕との関係において最も接近した2点間の距離、即ち、ロボット間の最短距離が最短距離記憶レジスタRmin.に保存されることとなる。

【0015】以上の処理により現時点におけるロボット間の最短距離Rmin.を算出したCPUは、最短距離Rmin.と予めプログラム設定された係数算出のための関数式fに基いて係数f(Rmin.)を算出し、この係数f(Rmin.)を不揮発性RAMに記憶されている手動送り速度

の設定値V0に乘じて送り速度 $V0 \cdot f(Rmin.) = V$ を求め(ステップS13)、教示操作盤Wで指定された産業用ロボットA1またはA2の各軸に1周期分のパルス分配を実行して送り速度Vでの手動送りを実行すると共に(ステップS14)、前述のシミュレーション機能により産業用ロボットA1、A2の各々の現在位置姿勢に基いて特徴点Pi、Qjの現在位置座標の全てを算出し、ロボットコントローラBの不揮発性RAMに更新記憶する。図3はROMにプログラム設定された係数算出のための関数式fの一例を示す図で、この関数式fはオーバーライド値100%を漸近線とし、最短距離Rmin.の減少に伴って係数f(Rmin.)の値が加速度的に減少する対数関数状の単調増加関数であって、Rmin.の値が“0”となっても係数(Rmin.)の値はオーバーライド値10%前後に保持されている。

【0016】このようにして1周期分の「手動送り処理」を終了したCPUは、オペレータ操作に基く教示操作盤Wからの手動送り指令が検出されなくなるまでの間、前述の「手動送り処理」を繰り返し実行し、産業用ロボットA1、A2間の最短距離Rmin.の値が小さくなれば図3の関数式fに基いて手動送り速度Vを自動的に低速化して衝突発生時のダメージを未然に防止し、また、産業用ロボットA1、A2間の最短距離Rmin.の値が大きくなって接触や衝突の心配がなくなれば、図3の関数式fに基いて手動送り速度Vを自動的に高速化して効率良く手動送りを行う。従って、従来のように産業用ロボットA1、A2間の位置関係が変化する毎にオペレータが安全性や作業効率等を考慮してオーバーライド値を一々切り替える必要はなく、作業効率が各段に向上する。例えば、産業用ロボットA1およびA2を共に教示モードとして2つのロボットの教示操作を行うときに、最初に産業用ロボットA1の位置姿勢を手動送りで変化させて産業用ロボットA2に接近させた状態で産業用ロボットA1に位置決め位置を教示してから産業用ロボットA2を手動送りして産業用ロボットA1に接近させ、再び、産業用ロボットA1に手動送りを施すような場合、新たに産業用ロボットA1の手動送りを開始する段階では最初に産業用ロボットA1に位置決め位置を教示したときに比べて産業用ロボットA1-A2間の距離が接近しているので、ロボットの接触や衝突を回避するためには産業用ロボットA1の送り速度を小さな値に再設定してから送り操作を開始する必要がある。従来このような再設定操作を行うためにはオペレータが教示操作盤Wを手動操作して送り速度の基準値となる設定値V0の新たな値を選択し、この値を不揮発性RAMに改めて設定しなければならなかったが、本実施例では手動送り操作により産業用ロボットA1、A2の位置姿勢が変化する毎に前述の「手動送り処理」で手動送り速度Vが自動的に再設定されるので、オーバーライド値の再設定等といった煩わしさは全くない。なお、手動送り速度の指令

値となる設定値 V_0 はシステムパラメータとして不揮発性RAMに予め設定されているが、従来と同様、教示操作盤Wからの指令により任意に設定変更することができる。

【0017】この実施例では、産業用ロボットA1、A2の各リンクやサーボモータ等の突出部材およびエフェクタの輪郭形状を端的に示す位置に一定の安全を見込んで幾つかの特徴点 P_i 、 Q_j を定義しているの、産業用ロボットA1、A2間の最短距離 R_{min} の値が“0”となつて計算上で接触または衝突状態と見做された場合であっても、実際には産業用ロボットA1、A2間に一定の間隙がある場合もある。最短距離 R_{min} の計算値が“0”となつた状態でも図3に示されるように係数 f (R_{min})の値はオーバーライド値10%前後に保持されているので、この状態から産業用ロボットA1、A2を微小速度 V で手動送りし、産業用ロボットA1、A2を更に接近させた状態で教示操作を行うことも可能である。この場合、実際の手動送り速度 V は極めて小さな値となっているから産業用ロボットA1、A2が実際に接触したり衝突したりしてもダメージを生じることはない。

【0018】また、前述の実施例のように係数算出のための関数式 f をプログラム設定して係数 f (R_{min})を算出する代わりに、最短距離 R_{min} の値、即ち、図3の横軸を一定の刻み幅で分割し、その各々に対応して係数 f (R_{min})の値を記憶させたファイルをROMに記憶しておいて、前述の「手動送り処理」で最短距離記憶レジスタに最終的に保存された最短距離 R_{min} が属する刻み幅の区分をステップS13の処理で検索し、これに対応する係数 f (R_{min})の値を検出して係数として用い

るようにしても良い。

【0019】また、最短距離 R_{min} の値が設定距離 d 以下のときにのみ実際の手動送り速度 V を手動送り速度の設定値 V_0 よりも低速にする場合には、図3に示されるような関数式 f に代えて図4に示されるような関数式 f を用いる。図4の関数式 f は $R_{min} \leq d$ の範囲を短調増加の線形関数 g で構成され、かつ、 $d < R_{min}$ の範囲を構成する関数 g' は f (R_{min})の値が常に一定(オーバーライド100%)である。そこで、このような関数式 f を用いる場合は、前述の「手動送り処理」におけるステップS13の処理で、まず、最短距離 R_{min} と設定距離 d との大小関係を比較し、 $R_{min} \leq d$ の場合には短調増加の線形関数の式 g で係数 f (R_{min})の値を算出する一方、 $d < R_{min}$ となつた場合には係数 f (R_{min})としてオーバーライド100%を無条件に出力し、 $V_0 \cdot f$ (R_{min}) = V の演算式を実行するようにする。この処理をファイルを用いて行うことも前述の場合と同様に可能である。

【0020】以上、産業用ロボットA1、A2の各リンクやサーボモータ等の突出部材およびエフェクタU1、

U2に幾つかの特徴点 P_i 、 Q_j を定義して「手動送り処理」を実施する場合を例に取って説明したが、産業用ロボットA1、A2やエフェクタU1、U2の輪郭形状を単純形状のソリッドモデル、例えば、球、円柱、直方体やその組み合わせに近似して産業用ロボットA1、A2間の最短距離 R_{min} を算出したり、また、産業用ロボットA1、A2およびエフェクタU1、U2の形状をB-repsやCSG等の完全な形の形状データに変換して前述のシミュレーション機能で各部の厳密な現在位置姿勢を得ることによって産業用ロボットA1、A2間の最短距離 R_{min} を算出することも可能である。この場合、産業用ロボットA1、A2やエフェクタU1、U2の形状を示すデータは大幅に増大することとなるが、最短距離 R_{min} を算出するためのアルゴリズム自体は前述の「手動送り処理」におけるステップS1～ステップS12までの処理と同様である。B-repsやCSG等の形状データで産業用ロボットA1、A2やエフェクタU1、U2の位置姿勢を算出する場合には最短距離 R_{min} の計算結果が実際の位置姿勢と良く一致し、最短距離 R_{min} の計算結果が“0”となつたときには特徴点を用いて計算を行った場合に比べて産業用ロボットA1、A2が更に接近することになるので、 R_{min} の計算結果が“0”のときの f (R_{min})の値が更に小さな値となるように関数式を設定し、衝突発生時のダメージを小さくするようにする。

【0021】更に、図5に示されるように単一のロボットコントローラBで3台以上の産業用ロボットA1、A2・・・Anを駆動制御する場合も、前述の「手動送り処理」と略同等のアルゴリズムを有する図9および図10に示すような「手動送り処理」の処理操作で各産業用ロボットの手動送り速度を自動的に調整することができる。以下、図9および図10のフローチャートを参照して3台以上の産業用ロボットA1、A2・・・Anを駆動制御する場合の処理動作を簡単に説明する。また、この実施例においてはロボットコントローラBの不揮発性RAMに図11に示されるようなファイルが設けられており、ユーザーがロボットの設置を完了した時点で、産業用ロボットA1、A2・・・Anの各々のコード番号に対応して干渉対象となるロボットのコード番号を設定できるようになっている。

【0022】教示操作盤Wからの手動送り指令を検出したCPUは、まず、最短距離記憶レジスタ R_{min} にシステムが許容する設定可能最大値を初期設定して検索指標 i の値を“0”に初期化した後(ステップT1、ステップT2)、該指標 i を“1”インクリメントする(ステップT3)。次いで、CPUは検索指標 i の値に基いて現時点で手動送り指令を入力されている産業用ロボット A_m ($m=1 \sim n$ のいずれか)の第 i 番目の特徴点 P_i の現在位置座標を不揮発性RAMから読み込んで一時記憶すると共に(ステップT4)、ロボット検索指標 k および検索指標 j の値を“0”に初期化し(ステップT

10

20

30

40

50

5, ステップT6)、ロボット検索指標 k を“1”インクリメントして(ステップT7)、該指標 k の値が m と一致するか否か、即ち、特徴点の現在位置座標データの読み込み対象となっている産業用ロボット A_k が現時点で手動送り指令を入力されている産業用ロボット A_m それ自体であるか否かを判別する(ステップT8)。そして、ステップT8の判別結果が偽となり、データの読み込み対象となっている産業用ロボット A_k が手動送り対象の産業用ロボット A_m でないことが確認されると、CPUは、更に、データの読み込み対象となっている産業用ロボット A_k が手動送り対象の産業用ロボット A_m と干渉する可能性のあるロボットであるか否かを判別することとなる(ステップT9)。即ち、データの読み込み対象となっている産業用ロボット A_k が現時点で手動送りの対象となっている産業用ロボット A_m に対応して図11のファイル手段に記憶されていればステップT9の判別結果は真、それ以外の場合は偽である。

【0023】そして、ステップT8の判別結果が偽となって、かつ、ステップT9の判別結果が真となった場合においてのみ、CPUは、前述の「手動送り処理」におけるステップS6～ステップS11の処理と同様にしてステップT10～ステップT15までのループ状の処理を繰り返し実行し、現時点で手動送り指令を入力されている産業用ロボット A_m およびスポット溶接器 U_m の特徴点 P_i [$i=1$]と産業用ロボット A_k およびスポット溶接器 U_k の特徴点 Q_j [$j=1$ ～最終値]との関係において最も接近した2点間の距離、即ち、産業用ロボット A_m およびスポット溶接器 U_m の特徴点 P_i [$i=1$]から最も近い位置にある産業用ロボット A_k およびスポット溶接器 U_k の特徴点までの距離を最短距離記憶レジスタ R_{min} に保存する。

【0024】次いで、CPUは、ロボット検索指標 k の値がロボット総数 n に達しているか否かを判別するが(ステップT16)、達していなければ再びステップT6の処理に復帰し、ロボット検索指標 k の値がロボット総数 n に達するまでの間、前述と同様の処理を繰り返し実行して、産業用ロボット A_m の干渉対象となる全てのロボットの全ての特徴点に対し、現時点で手動送り指令を入力されている産業用ロボット A_m およびスポット溶接器 U_m の特徴点 P_i [$i=1$]までの距離を算出し、その最小値を最短距離記憶レジスタ R_{min} に保存する。なお、ステップT8の判別結果が真となった場合もしくはステップT9の判別結果が偽となった場合にはステップT10～ステップT15までの処理を非実行とし、産業用ロボット A_m 自体の特徴点が重複して計算されることにより特徴点間距離が“0”となるのを防止すると共に、産業用ロボット A_m に干渉する可能性のないロボットに対して最短距離を求めるといった時間の無駄を排除する。

【0025】そして、このような処理を繰り返し実行す

る間にステップT16の判別結果が真となってロボット検索指標 k の値がロボット総数 n に達すると、最短距離記憶レジスタ R_{min} には、現時点で手動送り指令を入力されている産業用ロボット A_m およびスポット溶接器 U_m の特徴点 P_i [$i=1$]から最も近い位置に位置する産業用ロボットの特徴点までの距離が記憶されることとなる。

【0026】以下、CPUは、検索指標 i の値が最終値に達するまでの間、即ち、ステップT17の判別結果が真となるまでの間、検索指標 i の値を順次“1”インクリメントして前述の処理を繰り返し実行し、産業用ロボット A_m の特徴点と産業用ロボット A_m の干渉対象となる他の産業用ロボットの特徴点との間の距離を順次求め、最短距離 R_{min} よりも小さな値を有する特徴点間距離が算出される毎に、この値を最短距離記憶レジスタ R_{min} に更新記憶してゆく。

【0027】従って、ステップT17の判別結果が真となった段階では、現時点で手動送り指令を入力されている産業用ロボット A_m およびスポット溶接器 U_m の特徴点 P_i [$i=1$ ～最終値]と産業用ロボット $A_2, A_3 \dots A_n$ の各特徴点 Q_j [$j=1$ ～最終値]との関係において最も接近した2点間の距離、即ち、産業用ロボット A_m と該産業用ロボット A_m に最も近接する産業用ロボットとの間の最短距離が最短距離記憶レジスタ R_{min} に保存されることとなる。

【0028】以下、CPUは前述の「手動送り処理」におけるステップS13～ステップS14の処理と同様にしてステップT8～ステップT19までの処理を実施し、現時点で手動送り指令を入力されている産業用ロボット A_m の手動送り速度を調整する。作用および効果の点に関して前述の実施例と同様であり、各ステップの処理においては、前述の実施例と同様、既に説明した様々なバリエーションを適用することができる。

【0029】また、産業用ロボット $A_1, A_2 \dots A_n$ の各々を図6に示されるように個別のロボットコントローラ $B_1, B_2 \dots B_n$ で駆動制御する場合には、ロボットコントローラ $B_1, B_2 \dots B_n$ の各々を通信線で接続し、各々のロボット $A_1, A_2 \dots A_n$ の特徴点の現在位置座標をそれぞれのロボットコントローラ $B_1, B_2 \dots B_n$ で算出して各々のロボットコントローラの記憶手段 $F_1, F_2 \dots F_n$ に更新記憶するようにし、手動送り対象となるロボットのロボットコントローラに記憶手段 $F_1, F_2 \dots F_n$ の特徴点のデータを逐次読み込んで、手動送り対象となるロボットのロボットコントローラが特徴点間の距離を算出するようにすれば、図9および図10に示される処理と同様にして各産業用ロボットの手動送り速度を自動的に調整することができる。

【0030】また、図7に示されるように、産業用ロボット $A_1, A_2 \dots A_n$ の各々を個別のロボットコン

トローラ B 1, B 2 . . . B n で駆動制御すると共にセルコントローラ C を設けて全体の協調動作を行わせるようにした場合は、セルコントローラ C によりロボット A 1, A 2 . . . A n の特徴点間の距離データを記憶手段 F に逐次一括して記憶させ、手動送り対象となるロボットのロボットコントローラに記憶手段 F のデータを逐次読み込ませて図 9 および図 10 に示されるような処理を行わせることもできる。図 8 に示されるように独立したセルコントローラを設けず、ロボットコントローラ B 1, B 2 . . . B n のいずれか一つに記憶手段 F を配備して、記憶手段 F を配備したロボットコントローラを前述のセルコントローラ C と同様のサーバ手段として用いるようにした場合もこれと同様である。

【0031】

【発明の効果】本発明におけるロボットの手動送り方法によれば、ロボットの制御装置が手動送り指令の入力されたロボットと他のロボットの現在位置姿勢とに基いてロボット間の最短距離を自動的に求め、しかも、この最短距離が短くなればなるほど手動送り速度を自動的に低速化してロボットを手動送りするので、ロボットの位置姿勢変化によってロボット間の距離が変化した場合でも教示操作の対象となるロボットの送り速度をオペレータが再設定する必要がなく、しかも、常に安全かつ効率の良い速度でロボットを手動送りすることができる。

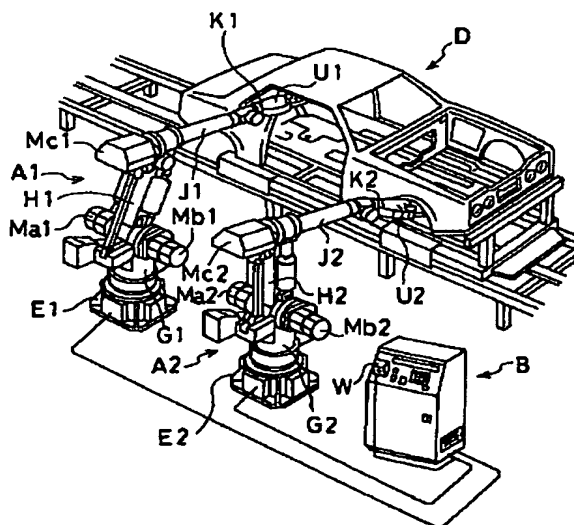
【図面の簡単な説明】

【図 1】相互に干渉する可能性のある領域に 2 台の産業用ロボットを配備した状態を示す斜視図である。

【図 2】本発明の方法を適用した一実施例の手動送り処理の概略を示すフローチャートである。

【図 3】同実施例の手動送り処理で係数を算出するために用いた単調増加関数の一例を示す概念図である。

【図 1】



【図 4】係数を算出するための単調増加関数の他の例を示す概念図である。

【図 5】単一のロボットコントローラで 3 台以上の産業用ロボットを駆動制御する場合を示すブロック図である。

【図 6】産業用ロボット毎のロボットコントローラで複数の産業用ロボットを駆動制御する場合を示すブロック図である。

【図 7】セルコントローラおよび産業用ロボット毎のロボットコントローラで複数の産業用ロボットを駆動制御する場合を示すブロック図である。

【図 8】産業用ロボット毎のロボットコントローラの一つをサーバとして複数の産業用ロボットを駆動制御する場合を示すブロック図である。

【図 9】本発明の方法を適用した別の実施例の手動送り処理の概略を示すフローチャートである。

【図 10】手動送り処理の概略を示すフローチャートの続きである。

【図 11】複数の産業用ロボットの各々のコード番号に対応して干渉対象となるロボットのコードを記憶したファイルを概念的に示す図である。

【符号の説明】

A 1 ~ A n 産業用ロボット

B, B 1 ~ B n ロボットコントローラ

C セルコントローラ

G 1, G 2 ショルダー (リンク)

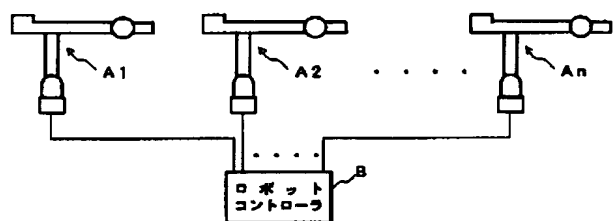
H 1, H 2 アーム (リンク)

J 1, J 2 フォアアーム (リンク)

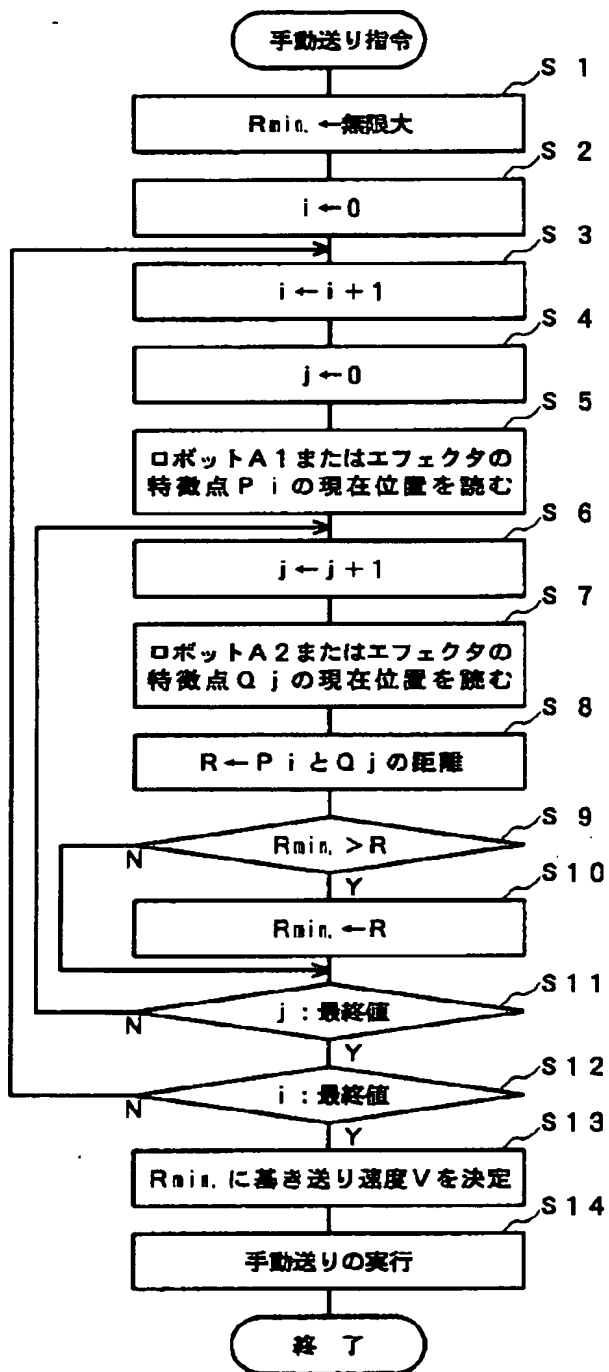
K 1, K 2 リスト (リンク)

U 1, U 2 スポット溶接器 (エフェクタ)

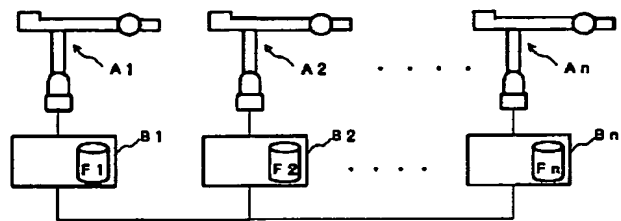
【図 5】



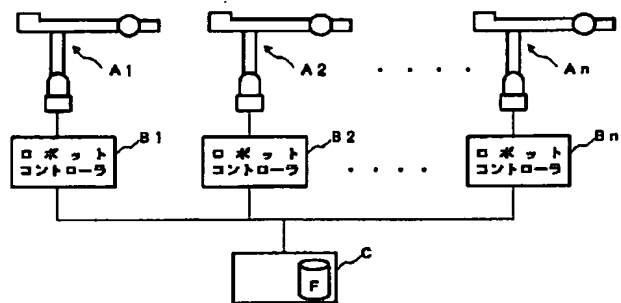
【図2】



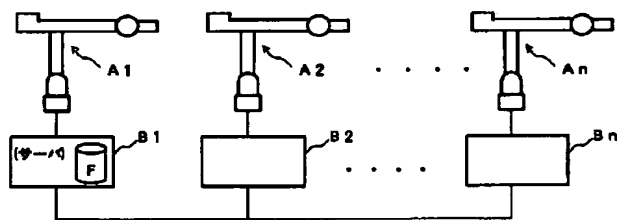
【図6】



【図7】



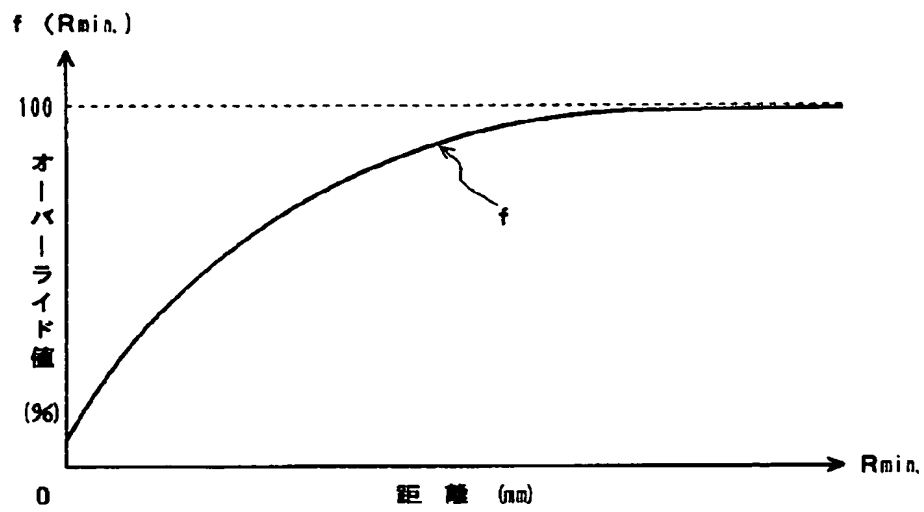
【図8】



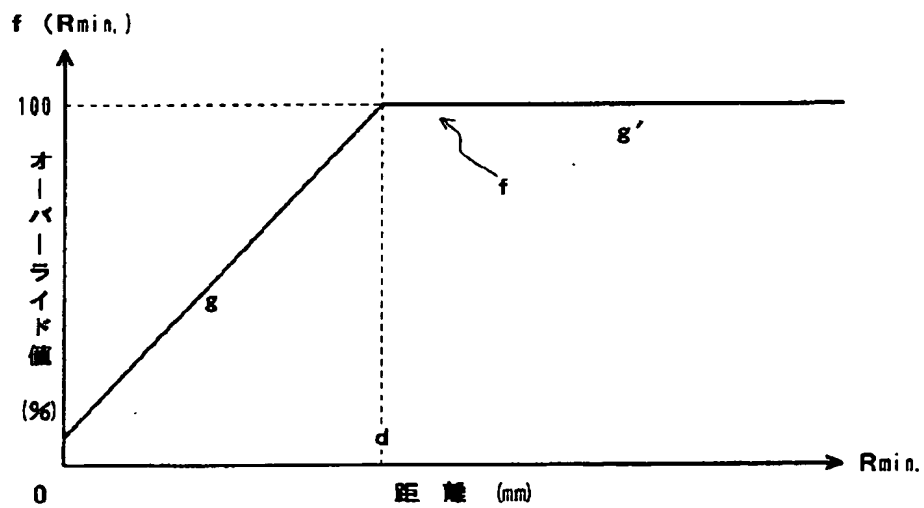
【図11】

ロボット名	干渉対象ロボット名
A1	A2, A3, A4
A2	A1, A3, A5
⋮	⋮
Ak	Ak-1, Ak+3, Ak+5
⋮	⋮
An	An-1, An-2

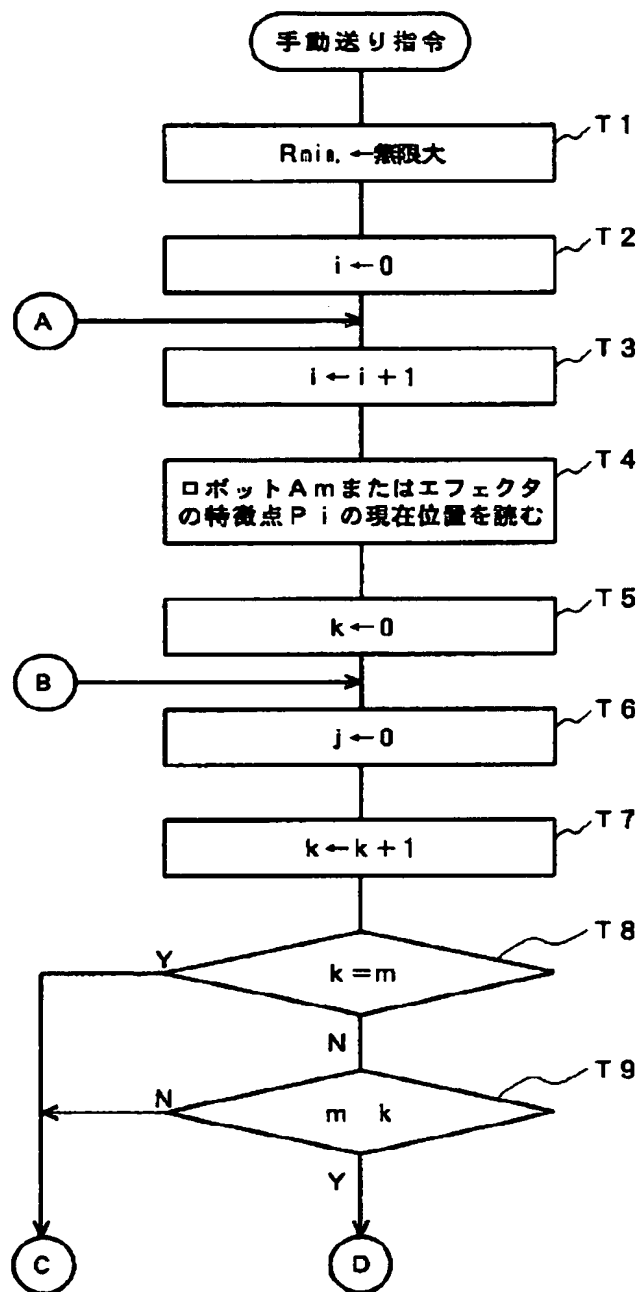
【図3】



【図4】



【図 9】



【図10】

